

What is claimed is:

1        1.     A method of loading a trustable operating system comprising:  
2                identifying a region in a memory of a computer by a one of a plurality of  
3        processors;  
4                loading a content into the region;  
5                registering an identity of the content of the secured region; and  
6                causing the one processor to jump to a known entry point in the content.

1        2.     The method of claim 1, further comprising:  
2                preventing interference with the identifying, loading, and registering by each of a  
3        remaining one of the plurality of processors.

1        3.     The method of claim 2, wherein preventing interference comprises halting  
2        each of the remaining ones of the plurality of processors until the identifying, loading,  
3        and registering is complete.

1        4.     The method of claim 2, further comprising:  
2                causing each of the remaining ones of the plurality of processors to jump to the  
3        known entry point in the content.

1        5.     The method of claim 1, wherein identifying comprises receiving a region  
2        parameter, the region parameter specifying a location of the region.

1        6.     The method of claim 5, wherein the location is a range of addresses in the  
2        memory of the computer within which the region is located.

1           7.       The method of claim 5, wherein the location comprises a start address and  
2       a length of the memory of the computer within which the region is located.

1           8.       The method of claim 1, wherein registering comprises:  
2       recording a hash digest of the content of the secured region; and  
3       signing the hash digest, the signed hash digest being stored in a register in the  
4       memory of the computer.

1           9.       The method of claim 1 wherein the content is a component of an operating  
2       system to operate the computer.

1           10.      The method of claim 9, wherein the operating system is one of a  
2       Windows operating system, a Windows 95 operating system, a Windows 98 operating  
3       system, a Windows NT operating system, a Windows 2000 operating system, a virtual  
4       machine monitor, and a privileged software nucleus.

1           11.      The method of claim 1 wherein identifying, loading and registering are  
2       uninterruptible.

1           12.      A article of manufacture comprising:  
2       a machine-accessible medium including a data that, when accessed by a machine  
3       cause the machine to,  
4       halt all but one of a plurality of central processing units (CPU) in a computer;  
5       identify a region in a memory of the computer;  
6       block access to the identified region by all resources except the non-halted CPU;  
7       load a content into the identified region;  
8       record a cryptographic hash of the content in the identified region; and

9 cause the non-halted CPU to begin executing at a known entry point in the  
10 identified region.

1 13. The article of manufacture of claim 12, wherein the data that causes the  
2 machine to halt the all but one of a plurality of CPUs comprises data causing the all but  
3 one of a plurality of CPUs to enter a halted state.

1 14. The article of manufacture of claim 13, wherein the data further causes the  
2 halted CPUs to exit the halted state after the non-halted CPU has begun executing at the  
3 known entry point in the identified region

1 15. The article of manufacture of claim 14, wherein the data further causes the  
2 previously halted CPUs to begin executing at the known entry point in the identified  
3 region upon exiting the halted state.

1 16. The article of manufacture of claim 13, wherein the data that causes the  
2 machine to record the cryptographic hash includes data that further causes the machine  
3 to,

4 erase a hash digest area in the memory of the computer;  
5 record a required platform information in the hash digest area;  
6 compute the cryptographic hash of the identified region; and  
7 record the computed cryptographic hash in the hash digest area.

1 17. The article of manufacture of claim 16, wherein the hash digest area is a  
2 register in the memory of the computer.

1        18. The article of manufacture of claim 13, wherein the data that causes the  
2 machine to identify the region in memory of the computer includes data that further  
3 causes the machine to receive at least one region parameter containing a location of the  
4 identified region.

1        19. The article of manufacture of claim 13, wherein the location includes an  
2 address of the identified region.

1        20. The article of manufacture of claim 13, wherein the location includes a  
2 length of the identified region.

1        21. A method of securing a region in a memory of a computer comprising:  
2            halting all but one of a plurality of central processing units (CPU) in a computer;  
3            blocking access to a region in a memory of the computer by all resources except  
4            the non-halted CPU;  
5            recording a cryptographic hash of the region; and  
6            placing the non-halted CPU into a known privileged state.

1        22. The method of claim 21, further comprising causing the non-halted CPU  
2 to jump to a known entry point in the region.

1        23. The method of claim 21, wherein halting comprises causing the all but one  
2 of a plurality of CPUs to enter a special halted state.

1        24. The method of claim 23, further comprising causing the halted CPUs to  
2 exit the special halted state after the non-halted CPU has been placed into the known  
3 privileged state.

1           25.    The method of claim 24, further comprising causing the previously halted  
2   CPUs to begin executing at a known entry point in the region upon exiting the special  
3   halted state.

1           26.    The method of claim 21, wherein recording the cryptographic hash  
2   comprises:

3           erasing a hash digest area in the memory of the computer; and  
4           recording a required platform information in the hash digest area;  
5           computing the cryptographic hash of the region's contents; and  
6           recording the computed cryptographic hash in the hash digest area.

1           27.    The method of claim 26, wherein the hash digest area is a register in the  
2   memory of the computer.

1           28.    The method of claim 26, wherein computing the cryptographic hash of the  
2   region's contents is performed by a digest signing engine coupled to the memory of the  
3   computer.

1           29.    The method of claim 21, wherein the region is specified in at least one  
2   region parameter.

1           30.    The method of claim 29, wherein the at least one region parameter is an  
2   address of the region in the memory of the computer that is to be secured.

1           31.    The method of claim 29, wherein the at least one region parameter is a  
2   length of the region in the memory of the computer that is to be secured.

1           32. An apparatus to load a trustable operating system comprising:  
2            a first processor having a start secure operation (SSO), the SSO having a memory  
3   region parameter, wherein the first processor executes the SSO to block access to a  
4   region of memory specified in the memory region parameter and to place a content in the  
5   specified region;  
6            a hash digest, wherein the first processor further executes the SSO to erase a  
7   current content of the hash digest and to record in the hash digest a cryptographic hash of  
8   the content of the specified region; and  
9            wherein the first processor further executes the SSO to unblock access to the  
10   specified region and to jump to a known entry point in the content of the specified region.

1           33. The apparatus of claim 32, further comprising:  
2            a second processor, the second processor having a join secure operation (JSO),  
3   wherein the second processor executes the JSO to prevent the second processor from  
4   interfering with the first processor's execution of the SSO.

1           34. The apparatus of claim 33, wherein the second processor commences  
2   execution of the JSO when the first processor commences execution of the SSO.

1           35. The apparatus of claim 33, wherein, to prevent the second processor from  
2   interfering with the first processor's execution of the SSO, the JSO causes the second  
3   processor to enter a halted state until the first processor's execution of the SSO is  
4   complete.

1           36. The apparatus of claim 35, wherein the first processor executes the JSO to  
2   further cause the second processor to exit the halted state after the first processor's

3 execution of the SSO is complete and to begin executing at the known entry point in the  
4 content of the specified region.

1 37. The apparatus of claim 32, further comprising a digest signing engine  
2 having a secure channel to access the hash digest, the digest signing engine computing  
3 the cryptographic hash of the content in the specified region in response to a request by  
4 the first processor executing the SSO.

1 38. The apparatus of claim 32, wherein the hash digest is a register in a  
2 memory of the apparatus outside the specified region.

10000000000000000000000000000000